IEOR242 Group6 Project

Statistical Analysis on Music Influence Relation with Neural Network

Name	**Student ID**
Malcolm Zhao	3039679554
Siyi Wu	3039679411
Lingxin Li	3039713458
Zihao Yang	3036781148
Zongxin Chen	3039713614
Wenzhe Gao	3034619953
Zhihao Du	3034708847

Motivation

Understanding the Significance of Musical Influence

The essence of this project is rooted in the dynamic and intricate tapestry of the music industry. Music, an art form that transcends boundaries and epochs, serves as a universal language, shaping cultures and personal identities. Within this realm, the concept of musical influence is particularly fascinating. It involves understanding how one artist might inspire, shape, or contribute to the development of another artist's musical style, composition, and overall artistry.

The motivation behind this project is twofold. Firstly, it seeks to unravel the intricate web of influences in the music world. By doing so, we can gain insights into how music evolves, how genres are formed, and how artists impact one another across time and space. Secondly, this analysis is pivotal for preserving and appreciating musical heritage. Understanding these influences allows us to trace the lineage of musical styles and appreciate the interconnectedness of artists across different eras.

The music industry, constantly evolving with technological advancements and shifting cultural tides, presents a fertile ground for applying data science. The project aims to bridge the gap between musicology and data analytics, employing the latter to explore and quantify a concept that has been traditionally qualitative and often subjective.

Data Manipulation

We have two datasets (Appendix A.1, A.2) containing detailed information about music artists and their songs produced in the 20th and 21st centuries. Artists are also being analyzed by their influence on other artists, as in an influencer with many followers. The details of the two datasets can be found in Appendix A Section. Given the musical features of each song and the relationships between artists, we are able to perform sophisticated analysis to study and understand how musical influence is imposed between artists. The following sections will go over data integration, cleaning, transformation, and generation processes during the study before building the model.

Data Integration

Key information for the analysis is located in both datasets. Thus, merging is required. Specifically, the *genre* attribute is essential information about a piece of music in the influence dataset. We performed a natural join on the *influencer_id* and *follower_id* of the influencer table on the *artist_id* of the full music table.

Data Cleaning

After a brief exploratory data analysis on our current dataset, we observed that there exist low-quality entries that require further attention. We deleted rows with missing values, and all the entries about songs before 1943. It is observed that there is an explicit cutoff in the quantity of entries collected before 1943. Thus, it was necessary to remove them to maintain the quality of our analysis.

Data Transformation

To fit into a neural network model, standardization of the numerical features of songs is required. We also replaced song names with unique identification to avoid confusion when duplicate song names exist.

Input Data Generation

As the final step in the data manipulation process, input data to the RNN model needs to be generated. Since the main objective is to determine whether an influencing relationship occurs between two artists, the input of the model will be concatenated representations of artists given by the features of their music. To generate these *song features* for each artist, 20 songs in chronological order are randomly selected for each unique artist in our dataset. This will be the basis of our input to the model.

After generating the *song features*, we need to create pairs of artists since the input to the model will be a pair of artists. In total, roughly 100,000 pairs of artists are generated. Among them, 50% have artistic influence, while the others do not.

Model

Why do we use RNN enhanced with LSTM layers?

When assessing whether an artist influences another, leveraging a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) layers is particularly effective. RNNs are good at processing the sequential nature of music, analyzing and retaining the progression of song elements like melody and rhythm. This memory function is important for recognizing how the traits of one song may impact subsequent creations, similar to an artist's influence over time.

LSTMs enhance this by preserving information over extended periods, which is crucial for capturing the subtle, long-range impact one artist may have on another's work. By representing an artist with features from 20 consecutive songs, LSTMs address the challenge of traditional RNNs that struggle with long-term dependencies, especially when song influences are not immediately successive. Gate Units within LSTMs manage to keep track of relevant information across these songs, ensuring the model discerns influence patterns even when they are not linear or directly connected, thus offering a nuanced analysis of artistic impact.

RNN Model Architecture and Application

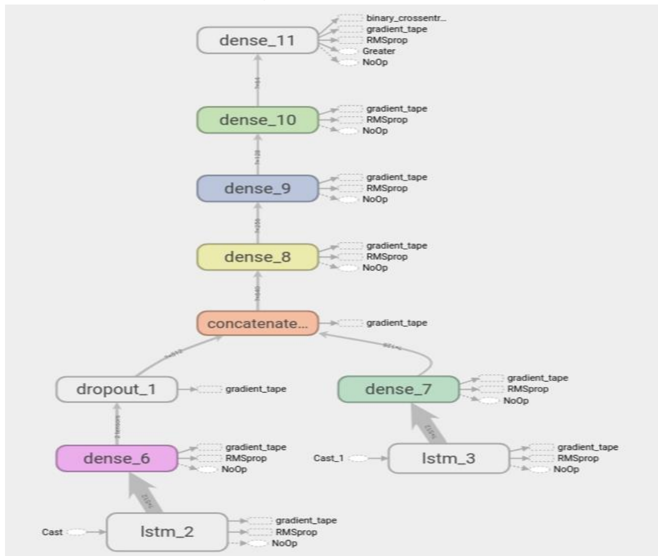
Our approach integrated these datasets into a sophisticated dual-input RNN model, designed to capture and analyze the nuances of musical influence.

Data Integration and Preprocessing:

- The Full Music Data was utilized to construct detailed profiles for each artist and track, focusing on attributes like tempo, energy, and danceability.
- The Influence Data was used to map the relationships between artists, as well as to identify influencers' and followers' music production genres.
- The data set is divided into a training set, validation set, and test set according to the ratio of 7: 1.5: 1.5.

Model Architecture (see Tensorboard Graph below):

- Influencer and follower data streams were processed through LSTM layers, each with 512 units, tailored to handle the sequential nature of the influence data, each with a dropout rate of 0.3 to handle the overfitting issue.
- Parallel dense layers (512 and 128 units respectively) are stacked to the two LSTM streams and the outputs are concatenated together after different dropout operations.
- Dense layers with varying units (256, 128, 64) follow the Concatenation layers, leading to a sigmoid activation function for binary classification.

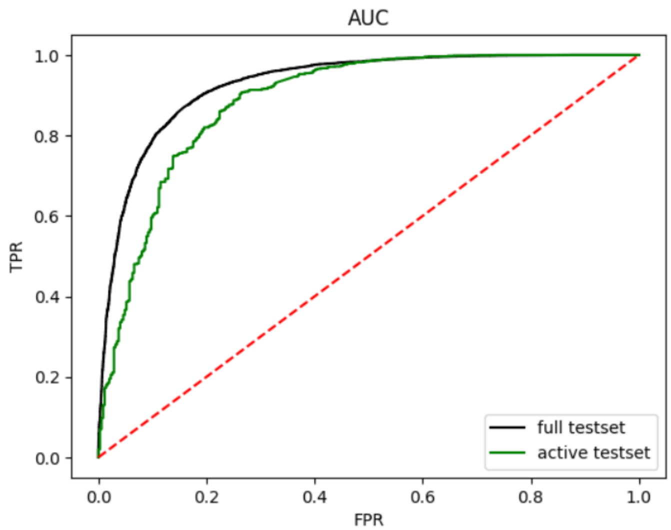


Optimization, Compilation, and Packages

- Optimization (RMSprop, learning rate: 0.01): Chosen for its efficiency in handling the vanishing gradient problem common in RNNs.
- Loss Function (Binary cross-entropy): Appropriate for binary classification tasks, aligning with our objective of predicting influence patterns.
- Metrics (Accuracy): Provides a straightforward measure of the model's performance in classifying influence relationships.
- Key Packages (Keras): Selected for its user-friendly interface and flexibility in building and experimenting with neural network architectures.

Results and Performance

Our model has done well in predicting the influence one artist may have over another. By analyzing selected features from the comprehensive datasets, the model reached an accuracy of 85.51% and an impressive AUC (Area Under the Curve) score of 0.9286. These results were derived from the test set. The AUC score, being close to 1, indicates a high true positive rate and a low false positive rate, which indicates the model's strong capability in discerning influential relationships.



Confidence in the Model

The robustness of the model is reflected in the implementation of monitoring functions such as EarlyStopping and ModelCheckpoint. EarlyStopping ceased training at epoch 34, preventing overfitting, while ModelCheckpoint identified the optimal model performance at epoch 27. This strategic use of callbacks ensured that we maintained high model performance without overtraining, enhancing our confidence in its generalizability. The confusion matrix further supports this, especially when looking at active artists (artists creating more than 19 music pieces), where the model achieved a higher accuracy of 87.85%, suggesting that our model is particularly adept at understanding the nuances of musical influence given sufficient music features, which is typical in the modern music industry.

Conclusion

In conclusion, the project has not only successfully demonstrated the capacity to predict musical influence with high accuracy but also presented direct applications for historical music analysis. For example, the model can discern if a Romantic-era composer was influenced by Bach, showcasing the potential to teach computers a deep understanding of musicology. The high accuracy and AUC scores underscore the model's effectiveness, making it a powerful tool for both historical music analysis and contemporary influence prediction.

Impact

Contributions and Future Potential

The project has made significant strides in quantifying the influence of one artist over another. With a prediction accuracy of 0.8551 and an AUC of 0.9286, the model demonstrates high reliability in assessing musical influences. This accomplishment has several impactful implications:

1. Educational Tool in Musicology: The ability to computationally determine if one artist is influenced by another, especially in classical music, provides an innovative tool for education. It enables learners to visually and quantitatively understand the flow of musical influence, making the study of music history and theory more interactive and data-driven.

2. Enhancing Music Discovery and Curation: For music platforms like Spotify, Apple Music, and others, this model can revolutionize how music is recommended and curated. By understanding an artist's influences, these platforms can create more nuanced and personalized playlists, catering to the sophisticated tastes of diverse listeners.

3. Cultural Preservation and Analysis: This approach offers a novel cultural and historical music analysis method. By mapping out influences, we can preserve the legacy of artists and understand cultural shifts in music over time.

Pathways for Enhancement

There are several avenues for further enriching this project:

1. Model Optimization: Exploring different parameters or a varied number of songs to represent an artist could refine the model's predictive accuracy.

2. Extending the Model's Application: Adapting the model to assess similarities in listener preferences could lead to more sophisticated user profiling and recommendation systems. This would not only cater to individual tastes but also explore communal trends in music listening.

3. Personalized User Experience in Music Platforms: Users' behavior could be viewed as a matrix representing the follower’s features. The influence model could thus generate an influence matrix indicating each musician’s, genre’s, and album’s influence on the user. Integrating this model’s output to generate user-specific images and recommendations would greatly enhance the user experience on music streaming platforms. This personalization could extend beyond playlists to include concert recommendations, artist discovery, and even interactive educational content about music history and theory.

In conclusion, the project stands at the intersection of music and data science, offering substantial contributions to both fields. Its potential for educational, cultural, and commercial impact is vast, opening doors to a deeper, data-driven understanding and appreciation of music.

Appendices

A. Data

A.1 Music Data (98340 rows * 19 columns)

Name	Description
artist_name	The artist
artist_id	Unique identification number
danceability	How suitable a track is for dancing
energy	The perception of intensity and activity
valence	The musical positiveness
tempo	tempo in beats per minute
loudness	overall loudness in decibels
mode	An indication of modality
key	The estimated overall key
acousticness	A confidence measure of whether the track is acoustic
instrumentalness	Predicts whether a track contains no vocals
liveness	Detects the presence of an audience in a track
speechiness	Detects the presence of spoken words in a track
explicit	Detects explicit lyrics in a track
duration_ms	The duration of the track in milliseconds
popularity	The popularity of the track
year	The year of release of a track
release_date	The calendar date of release of a track
song_title	The name of the track

A.2 Influence Data (42770 rows * 8 columns)

Name	Description
influencer_id	The same unique identification number given in the full_music_data
influencer_name	The name of the influencing artist as given by the follower or industry experts
influencer_main_genre	The genre that best describes the bulk of the music produced by the influencing artist
influencer_active_start	The decade that the influencing artist began their music career
follower_id	A unique identification number given to the artist listed as a follower
follower_name	The name of the artist following an influencing artist

follower_main _genre	The genre that best describes the bulk of the music produced by the following artist
follower_active_start	The decade that the following artist began their music career

B. Codes and Data Source

B.1 Links

Relevant coding files and data can be found on the google drive.

B.2 Table of Content

File Name	**Purpose**
1.music_data_manipulation.ipynb	Python programs to preprocess data
2.musician_sequential_input_gen.ipynb	Python programs to generate model input data
3.musician_influence_with_RNN.ipynb	Python program for the RNN model
influence_musicians.h5	Final RNN model
data	Folder which stores the raw data
rearrange_data	Folder which stores the preprocessed data
variables	Folder which stores all the relevant variables generated by the Python library “pickle”
.ipynb_checkpoints	Cache file folder created by Jupyter Notebook